

```

Make_Wetmore_Plot <- function(Oral.Equiv,Exposure.Predictions=NULL,
dose.CAS.col="CASRN",
dose.lower.col="Lower.Oral.Equivalent.mg.kg.",
dose.name.col="Compound",
dose.order.col=NULL,
predictions.CAS.col="CAS",
predictions.lower.col="lcl",
predictions.upper.col="ucl",
predictions.mean.col="gm",
predictions.nearfield.col="NearField",
random.suppress=1,
log="y",
xlab="Compound",
ylab="mg/kg/day",
CAS.start=1,CAS.end=NULL,ylim=c(5*10^-2,5*10^3),cex.compound=0.5)
{
  axTexpr <- function(side, at = axTicks(side, axp=axp, usr=usr, log=log,cex.axis=cex.axis),
  axp = NULL, usr = NULL, log = NULL,cex.axis=1)

  {
    ## Purpose: Do "a 10^k" labeling instead of "a e<k>"
    ##      this auxiliary should return 'at' and 'label' (expression)
    ##
    ## Arguments: as for axTicks()
    ##
    ## Author: Martin Maechler, Date: 7 May 2004, 18:01
    eT <- floor(log10(abs(at)))# at == 0 case is dealt with below

    mT <- at / 10^eT
    ss <- lapply(seq(along = at),
    function(i) if(at[i] == 0) quote(0) else
    substitute(10^E, list(A=mT[i], E=eT[i])))

    do.call("expression", ss)
  }

  CAS.list <- unique(Oral.Equiv[,dose.CAS.col])
  if(is.null(CAS.end)) CAS.end <- length(CAS.list)

  if (is.null(dose.order.col)) dose.order.col <- dose.lower.col
  mean.OEs <- rep(Inf,length(CAS.list))
  for (this.CAS in CAS.list)
  {
    these.OEs <- Oral.Equiv[Oral.Equiv[,dose.CAS.col]==this.CAS,dose.order.col]
    mean.OEs[CAS.list==this.CAS] <- mean(these.OEs)
  }

  plot(-1,-1,xlim=c(0,1),ylim=ylim,log=log,xlab=xlab,ylab=ylab,xaxt="n",cex.lab=1.5,yaxt="n")
  aY <- axTicks(2)
  aY <- aY[log10(aY)==round(log10(aY))]
  axis(2, at=aY, label= axTexpr(1, aY),cex.axis=1.5)
  axis(1,at=1:length(CAS.list)/(length(CAS.list)+2),labels=FALSE)
}

```

```

name.list <- NULL

for (this.CAS in CAS.list[order(mean.OEs)][CAS.start:CAS.end])
{
  x.pos <-
  which(CAS.list[order(mean.OEs)][CAS.start:CAS.end]==this.CAS)/(length(CAS.list[order(mean.OEs)][CA
S.start:CAS.end])+2)

  these.OEs <- Oral.Equiv[Oral.Equiv[,dose.CAS.col]==this.CAS,dose.lower.col]
  matplot(rep(x.pos,length(these.OEs)),these.OEs,pch=19,cex=1,add=TRUE)

  name.list <- c(name.list,Oral.Equiv[Oral.Equiv[,dose.CAS.col]==this.CAS,dose.name.col][1])

  if (!is.null(Exposure.Predictions))
  {
    if (this.CAS %in% Exposure.Predictions[,predictions.CAS.col])
    {
      index <- Exposure.Predictions[,predictions.CAS.col]==this.CAS

      if (!is.null(predictions.nearfield.col))
      {
        if (Exposure.Predictions[index,predictions.nearfield.col]) line.col <-"red"
        else line.col <- "blue"
      } else line.col <- "green"

      if (!is.null(predictions.lower.col))
      lines(c(x.pos,x.pos),c(Exposure.Predictions[index,predictions.lower.col],Exposure.Predictions[index,predic
tions.upper.col]),lwd=1,col=line.col)
      else if (runif(1,0,1)<random.suppress)
      matplot(x.pos,Exposure.Predictions[index,predictions.mean.col],pch=15,cex=1,col=line.col,add=TRUE)
    }
  }
}
text((1+1:length(CAS.list))/(length(CAS.list)+2),10^(par("usr")[3] -
0.1),labels=name.list,cex=cex.compound,srt=45,pos=2,xpd=T)
}

```